EEL 6764 001 **Graduate Computer Architecture** **Midterm Exam**
Spring 2025 **SET A** **Your Name:**

IMPORTANT:
1) **Print this exam**
2) **Answer in your own hand-writing**
3) **Cut-n-paste from HW solutions will get you 0 points.**
4) **Scan and Upload in PDF format**

Notes:
- Closed Text and Closed Notes Exam
- Time: 5:00 pm – 6:15 pm.
- There are **5** questions. Answer all questions.
- Maximum points: 50.
- Answer in clear and legible handwriting. Partial credit will be given.

I	II	III	IV	V	Total

I. (10 pts.) Short questions, Fill in the blanks, True or False.

1. Some microprocessors today are designed to have adjustable voltage, so 15% reduction in voltage may result in 15% reduction in frequency. What is the impact on dynamic power? Show all your work. No credit if supporting work is not shown.

2. Higher associativity reduces the miss rate. True/False?

3. Why should we give priority to *read misses* over *write misses*?

4. You are designing a processor that spends 30% of time on data movement (loads/stores), 40% in ALU operations, 20% in control flow, and 10% in the rest. Given time to market constraint you know that only ALU operations can be sped up. To beat a competitor, your manager is demanding that you speed up the processor by 8X. Do you think **theoretically** this is possible? If so, how? If not, why not? No credit if you do not justify your answer.

5. *Execution time* is an accurate measure of computer performance. True/False.

6. Give an example where data forwarding *cannot* resolve a data hazard.

7. In today's processors **write-through** policy is widely used. Is this true? If not, explain why not?

8. Branches comprise 20% of all instructions. There is a 2-cycle stall (SP) on each mis-prediction. What should be the desired accuracy of the branch prediction scheme to obtain an average branch penalty of no more than 30% of ideal CPI.

9. In _____ scheme, on a miss, we fetch more than one block to reduce miss penalty or miss rate.
10. Circle appropriately: For way-predicting caches, the bandwidth is **better /worse/ no change**, power consumption is **better / worse**, hardware cost is **low / high / none**

II. (10 pts) Multiple choice questions. More than one choice may be correct. Credit only if you choose all correct choices.

1. Which of the following page replacement algorithm is popular in processors?
   (a) Random (b) LRU (c) Most Recently Used (d) None of the Above.
2. In an out-of-order processor, the following hazard(s) can occur?
   (a) RAW (b) WAW (c) WAR (d) None of the above.
3. The number of dies per 300 mm (30 cm) wafer for a die that is 1.5 cm on a side:
   (a) 370 (b) 150 (c) 270 (d) None of the Above.
4. In a multi-level cache say L1 and L2, our goal is to
   (a) reduce L1 hit time (b) reduce L1 miss rate
   (c) reduce L2 hit time (d) reduce L2 miss rate
5. Compared to 1-bit prediction scheme, which scheme(s) perform better?
   (a) 2-bit (b) Static (c) Correlating (d) All of the Above.
6. The advantage(s) of virtual memory are:
   (a) User does not have to worry about actual physical memory size.
   (b) User is given an illusion of an infinite memory.
   (c) Translation of virtual addresses incurs no latency.
   (d) All of the above.
7. Supply voltage scaling helps in improving:
   (a) Power (c) Performance
   (b) Energy (d) None of the above
8. The Write Back scheme can be improved by:
   (a) Write Buffer (b) Critical Word first
   (c) Dirty blocks (d) None of the above.
9. Which of the following can improve cache performance?
   (a) Larger block size (b) Higher Associativity
   (c) Multi-level Cache (d) None of the above
10. The main advantages of merging write buffer optimization is
    (a) reduces hit-time (b) reduces miss penalty
    (c) reduces miss rate (d) none of the above.

III. **Memory Hierarchy**

(**6 pts)** Briefly (in 2-3 sentences each) describe the following **four** advanced cache optimization techniques.

a) Nonblocking cache
b) Way predicting cache
c) Critical word first
d) Hardware prefetching

(4 pts) Consider the usage of critical word first and early restart on L2 cache misses. Assume a 1 MB L2 cache with 64-byte blocks and a refill path that is 16 bytes wide. Assume the L2 can be written with 16 bytes every 4 processor cycles, the time to receive the first 16 byte block from the memory controller is 120 cycles, each additional 16 byte block from main memory requires 16 cycles, and data can be bypassed directly into the read port of the L2 cache. Ignore any cycles to transfer the miss request to the L2 cache and the requested data to the L1 cache. How many cycles would it take to service an L2 cache miss with and without critical word first and early restart?

IV. Technology Trends, Performance Measurement

1. (4 pts) Describe briefly how the following ISA works?
   Stack, Accumulator, Register-memory, register-register. (No need of a figure)
2. (6 pts) We begin with a computer implemented in single-cycle implementation. When the stages are split by functionality, the stages do not require exactly the same amount of time. The original machine had a clock cycle time of 13 ns. After the stages were split, the measured times were IF, 2 ns; ID, 5 ns; EX, 1 ns; MEM, 4 ns; and WB, 2.5 ns. The pipeline register delay is 0.2 ns.
   a) What is the clock cycle time of the 5-stage pipelined machine?
   b) If there is a stall for every four instructions, what is the CPI of the new machine?
   c) What is the speedup of the pipelined machine over the single-cycle machine?
   d) If the pipelined machine had an infinite number of stages, what would its speedup be over the single-cycle machine? Assume ideal case of zero stalls.

V. Basic Pipelining and RISC-V architecture

A. Predicting which way branch will be taken statically (always taken or not taken) can give rise to mispredictions. This can be improved by **dynamic** prediction.

1. (4 pts) With the help of a state diagram explain how a **2-bit dynamic** prediction scheme works. Explain why the scheme works compared to a static scheme.
2. (2 pts) Can we do better than 2-bit prediction scheme? If so, how?

B. (4 pts) Answer the following RISC-V Base ISA architecture:

a) Name any *four ISA* design principles that RISC-V implements.

b) No. of integer registers _____ No. of fp registers _____

c) Instruction word length _____

d) No. of instruction formats _____

Formulas

1. $Energy \propto \frac{1}{2} \times Capacitive\ Load \times Voltage^2$
2. $Power \propto \frac{1}{2} \times Capacitive\ Load \times Voltage^2 \times Frequency\ Switched$
3. $Cost\ of\ Integrated\ Circuit = \frac{Cost\ of\ die + Cost\ of\ testing\ die + Cost\ of\ packaging\ and\ final\ test}{Final\ test\ yield}$
4. $Cost\ of\ Die = \frac{Cost\ of\ Wafer}{Dies\ per\ wafer \times Die\ Yield}$
5. $Dies\ per\ wafer = \frac{\pi \times (\frac{Wafer\ diameter}{2})^2}{Die\ Area} - \frac{\pi \times \text{Wafer Diameter}}{\sqrt{2 \times Die\ Area}}$
6. $Die\ Yield = Wafer\ Yield \times \frac{1}{(1 + Defects\ per\ unit\ area \times Die\ Area)^N}$
7. If X is 'n' times fast as Y then: $Execution\ Time(n) = \frac{Execution\ Time_Y}{Execution\ Time_X} = \frac{Performance_X}{Performance_Y}$
8. Amdahl's Law:
   $Speedup_{Overall} = \frac{Execution\ Time_{old}}{Execution\ Time_{new}} = \frac{1}{(1 - Fraction_{enchanced}) + \frac{Fraction_{enchanced}}{Speedup_{enchanced}}}$
9. $CPU\ Time = CPU\ Clock\ Cycles\ for\ a\ program \times Clock\ Cycle\ Time$
10. $CPI = \frac{CPU\ clock\ cycles\ for\ a\ program}{Instruction\ Count}$
11. $CPU\ Time = Instruction\ Count \times Cycles\ per\ Instruction \times Clock\ cycle\ Time$
12. $CPU\ Clock\ Cycles = \sum_{i=1}^{n}(IC_i \times CPI_i)$
13. $CPU\ Time = \sum_{i=1}^{n}(IC_i \times CPI_i) \times Clock\ Cycle\ Time$
14. $CPI = \frac{\sum_{i=1}^{n}(IC_i \times CPI_i)}{Instruction\ Count} = \sum_{i=1}^{n}(\frac{IC_i}{Instruction\ Count} \times CPI_i)$
15. $\frac{Misses}{Instruction} = Miss\ rate \times \frac{Memory\ accesses}{Instruction}$
16. $Average\ Memory\ Access\ Time = Hit\ Time + Miss\ Rate \times Miss\ Penalty$
17. For Multilevel Caches:
    $Average\ Memory\ Access\ Time = Hit\ Time_{L1} + Miss\ Rate_{L1} \times (Hit\ Time_{L2} + Miss\ Rate_{L2} \times Miss\ Penalty_{L1})$
18. $Memory\ Stall\ Cycles = Number\ of\ Misses \times Miss\ Penalty$
    $= IC \times \frac{Misses}{Instruction} \times Miss\ Penalty$
    $= IC \times \frac{Memory\ Accesses}{Instruction} \times Miss\ Rate \times Miss\ Penalty$
19. $Memory\ Stall\ Cycles = IC \times Reads\ per\ Instruction \times Read\ Miss\ Rate \times Rate\ Miss\ Penalty + IC \times Writes\ per\ Instruction \times Write\ Miss\ Rate \times Write\ Miss\ Penalty$

20. $CPU\ Execution\ Time = (\ CPU\ Clock\ Cycles + Memory\ Stall\ Cycles\ ) \times Clock\ Cycle\ Time$
21. $CPU\ Execution\ Time = IC \times \left(CPI_{execution} + \frac{Memory\ Stall\ Clock\ Cycles}{Instruction}\right) \times Clock\ Cycle\ Time$
22. $CPU\ Execution\ Time = IC \times \left(CPI_{execution} + \frac{Misses}{Instruction} \times Miss\ Penalty\right) \times Clock\ Cycle\ Time$
23. $CPU\ Execution\ Time = IC \times \left(CPI_{execution} + Miss\ Rate \times \frac{Memory\ Accesses}{Instruction} \times Miss\ Penalty\right) \times Clock\ Cycle\ Time$
24. $\frac{Memory\ stall\ Cycles}{Instruction} = \frac{Misses}{Instruction} \times (\ Total\ Miss\ Latency - Overlapped\ Miss\ Latency)$
25. $2^{index} = \frac{Cache\ Size}{Block\ size \times Set\ Associativity}$
26. $\frac{Memory\ stall\ Cycles}{Instruction} = \frac{Misses_{\ L1}}{Instruction} \times Hit\ Time_{L2} + \frac{Misses_{\ L2}}{Instruction} \times Miss\ Penalty_{L2}$
27. $Average\ Instruction\ Execution\ Time = Clock\ Cycle \times Average\ CPI$
28. $Speedup\ from\ Pipelining = \frac{Average\ instruction\ time\ unpipelined}{Average\ instruction\ time\ pipelined}$
29. $Speedup\ from\ Pipelining = \frac{CPI\ unpipelined \times Clock\ Cycle\ Unpipelined}{CPI\ pipelined \times Clock\ Cycle\ pipelined}$
30. $CPI\ Pipelined = Ideal\ CPI + Pipeline\ Stall\ Clock\ Cycles\ per\ Instruction = 1 + Pipeline\ stall\ cycles\ per\ Instruction$
31. $Speedup = \frac{CPI\ unpipelined}{1 + Pipeline\ stall\ cycles\ per\ Instruction}$
32. $Speedup = \frac{Pipeline\ depth}{1 + Pipeline\ stall\ cycles\ per\ Instruction}$
33. $Speedup\ from\ pipelining = \frac{1}{1 + Pipeline\ stall\ cycles\ per\ Instruction} \times \frac{Clock\ Cycle\ unpipelined}{Clock\ cycle\ pipelined}$
34. $Pipeline\ depth = \frac{Clock\ Cycle\ unpipelined}{Clock\ cycle\ pipelined}$
35. $Speedup\ from\ pipelinig = \frac{1}{1 + Pipeline\ stall\ cycles\ per\ Instruction} \times Pipeline\ depth$
36. $Average\ Instruction\ Time = CPI \times Clock\ Cycle\ Time$